STUDY AND DEVELOPMENT OF NETWORK BASED INTRUSION DETECTION SYSTEM

THESIS SYNOPSIS

SUBMITTED TO

Institute of Advanced Research, Koba Institutional Area, Gandhinagar 382426, Gujarat, India

Doctor of Philosophy

IN
COMPUTER SCIENCE

Submitted By
AZRIEL HENRY
(Reg. No.: IAR/11593)

UNDER THE GUIDANCE OF

PROF. SAMRAT KHANNA
DEPARTMENT OF COMPUTER SCIENCES & ENGINEERING
INSTITUTE OF ADVANCED RESEARCH,
GANDHINAGAR, GUJARAT, INDIA

Table of Contents

Content	Page No.
General Overview	3
Chapter 1 – Introduction	3
Chapter 2 – A Composite Approach of Intrusion Detection Systems: Hybrid RNN and Correlation-Based Feature Optimization	5
Chapter 3 – Hybrid CNN-GRU: A Deep Learning Technique for Intrusion Detection System	10
Chapter 4 – Assessing the Effectiveness of GRU for Deep Learning-Based Intrusion Detection	13
Chapter 5 – Enhancing Dataset for Optimized Classification in Intrusion Detection System	16
Chapter 6 – Conclusion & Future Scope	20
List of Publications and Conferences	22

GENERAL OVERVIEW

The escalating sophistication and prevalence of cyber threats pose a significant challenge to the security of modern networks. Intrusion Detection Systems (IDS) have emerged as critical components in safeguarding digital infrastructure by identifying and mitigating potential threats in real-time. Traditional IDS methods, particularly signature-based approaches, have demonstrated effectiveness in detecting known attack patterns. However, their inability to identify novel or evolving threats has necessitated the exploration of advanced anomaly-based detection techniques.

This thesis, titled "Study and Development of Network-Based Intrusion Detection System," focuses on leveraging advanced Deep Learning (DL) methodologies to enhance the detection capabilities of IDS. The research explores innovative solutions for addressing critical challenges, including class imbalance, high false-positive rates, and the complexity of feature selection. By integrating DL models such as Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN), this work aims to provide a robust framework for real-time intrusion detection.

A significant contribution of this study lies in the hybridization of CNN and GRU models, which demonstrated improved performance over existing approaches. Additionally, the application of the Synthetic Minority Oversampling Technique (SMOTE) effectively addressed the issue of imbalanced datasets, enabling comprehensive detection of diverse attack types. The research utilized the CICIDS 2017 dataset, known for its wide range of realistic attack scenarios, ensuring rigorous evaluation of the proposed methods. The findings underscore the transformative potential of combining deep learning with advanced data preprocessing techniques in the field of IDS. This work not only advances the theoretical understanding of DL applications in cybersecurity but also provides practical insights into the development of scalable and adaptive intrusion detection systems. The methodologies and outcomes presented in this thesis serve as a stepping stone for future research, highlighting the importance of continual innovation to counter the dynamic landscape of cyber threats.

CHAPTER 1: INTRODUCTION

The first chapter lays a comprehensive foundation for the thesis by reviewing the current state of cybersecurity and IDS technologies. It provides an in-depth analysis of the limitations of traditional methods and the opportunities presented by DL techniques. The chapter also outlines the research objectives and methodology, emphasizing the critical role of innovation in addressing evolving cyber threats.

Cybersecurity Landscape and the Role of Intrusion Detection Systems

The increasing prevalence and sophistication of cyber threats have made cybersecurity a critical concern for organizations worldwide. Intrusion Detection Systems (IDS) play an essential role in safeguarding digital infrastructures by identifying and mitigating unauthorized access attempts, malicious activities, and system anomalies. The primary objective of an IDS is to monitor network traffic and system behaviors, detecting deviations from expected patterns that may indicate cyber-attacks. IDS can broadly be classified into two types: host-based IDS,

which focus on individual systems, and network-based IDS, which analyze network-wide traffic. Traditional IDS rely heavily on signature-based detection methods, which match observed patterns with a database of known attack signatures. While effective in detecting previously documented threats, these methods struggle with identifying novel or sophisticated attacks that deviate from established patterns. Anomaly-based detection, on the other hand, identifies deviations from normal behaviors, enabling the detection of unknown threats. However, this approach often generates false positives, making it less reliable without further optimization.

Advancements in Intrusion Detection Techniques

Recent advancements in Machine Learning (ML) and Deep Learning (DL) have revolutionized IDS development by addressing the limitations of traditional methods. DL algorithms, such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), are particularly suited for analyzing the temporal and sequential nature of network traffic. These models can capture intricate patterns and dependencies within the data, enhancing detection accuracy and reducing false positives. Similarly, Convolutional Neural Networks (CNN) excel in feature extraction, providing valuable insights into complex network behaviors. The research presented in this thesis builds upon these advancements to propose innovative IDS frameworks. By integrating DL methodologies with data preprocessing techniques, such as the Synthetic Minority Oversampling Technique (SMOTE), the study aims to enhance detection capabilities and address challenges like class imbalance and high false-positive rates.

Motivation and Objectives of the Study

The motivation for this research stems from the critical need for robust and adaptive IDS capable of combating the dynamic nature of cyber threats. Traditional systems are often constrained by limited detection scopes, computational inefficiencies, and poor adaptability to emerging attack patterns. This study seeks to address these issues by:

1) To study and compare the current state of the approaches in the area of Anomaly-based intrusion detection
2) To design an Intrusion Detection System (IDS) using Deep Learning technique.
3) To evaluate the performance of the proposed technique using optimized IDS dataset and to compare the proposed model with existing methods.

Contributions of the Research

This thesis contributes significantly to the domain of IDS by introducing novel methodologies that improve upon existing detection frameworks. The research progresses through a series of stages, starting with the design and evaluation of RNN-LSTM-GRU-based IDS, followed by the development of a hybrid CNN-GRU model. The final stage incorporates SMOTE to address data imbalance, ensuring comprehensive detection across all attack types. The CICIDS 2017 dataset, a benchmark in IDS research, serves as the experimental foundation, providing diverse and realistic attack scenarios for evaluation.

Significance of the Research

The study's findings demonstrate the transformative potential of DL and advanced data preprocessing in enhancing IDS capabilities. By combining state-of-the-art algorithms with practical implementation strategies, this research provides a scalable and effective framework for real-time intrusion detection. The methodologies and insights presented in this thesis are expected to influence future developments in cybersecurity, paving the way for more resilient and adaptive IDS solutions. In summary, the introductory chapter establishes the importance of addressing the limitations of current IDS approaches through innovative research. It highlights the potential of integrating DL techniques with data balancing methods to create robust, scalable, and efficient intrusion detection systems capable of safeguarding critical infrastructures against the ever-evolving landscape of cyber threats.

CHAPTER 2: A COMPOSITE APPROACH OF INTRUSION DETECTION SYSTEMS: HYBRID RNN AND CORRELATION-BASED FEATURE OPTIMIZATION

Deep learning provides a scope to develop flexible IDSs. Some examples of deep learning algorithms are Deep Neural Network (DNN), Deep Boltzmann Machine (DBM), Deep Belief Network (DBN), Restricted Boltzmann Machine (RBM), Recur-rent Neural Network (RNN), etc. are used to generate labels for unlabelled data (Vinayakumar et al., 2019a) (Kwon et al., 2017b). Dataset plays has a key part to perform when using knowledge strategies including deep learning and machine learning models KDD 99 dataset, ICSI Enterprise Trac-ing Project, Australian Centre for Cyber Security published dataset - UNSW-NB15, CAIDA dataset, DARPA Lincoln Lap Packet trace, CIDDS-001, or CICIDS 2017. CI-CIDS 2017 contains a broad series of common attacks as well as benign. We have used a deep learning algorithm - RNN to develop an IDS. RNN has the ability to generate cycles of edges to form self-feeding a neuron's connections to itself over time. RNN introduces a time-step edge which makes it different from a feed-forward Neural Network. However, it has a longer dependencies issue which can be addressed using LSTM or GRU. The longer dependencies issue or long-term dependency arises when exponentially smaller weights are given to the long-term interactions. Hence it becomes difficult for the model to track the past information. LSTM works as a memory cell or a block to overcome the dependencies issue. GRU is the improved version the of LSTM framework with lesser gates. RNN-LSTM architecture is widely used in many applications.

Motivation

Intrusion detection systems (IDS) are critical in safeguarding digital infrastructures against a growing array of cyber threats. Despite significant advancements, existing IDS technologies, particularly anomaly-based systems, still struggle with achieving high detection accuracy while maintaining low false positive rates. These challenges are exacerbated by the evolving nature of cyber-attacks and the vast amount of data generated in modern networks. In our research study, the primary objectives were to study and compare current IDS approaches, develop a robust IDS using deep learning techniques, and evaluate its performance against optimized

datasets. Initially, I reviewed recent literature across several categories including ensemble learning-based IDS, federated learning-based IDS, statistical flow features-based IDS, and deep learning approaches. This review underscored the potential of deep learning methods in enhancing IDS performance, particularly in dealing with complex patterns and temporal dependencies inherent in network traffic data. To address these challenges, I developed an IDS using a hybrid model of Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). This approach, detailed in my published work, demonstrated high accuracy and reduced false positives with fewer features. However, it failed to detect certain attack types, highlighting a gap in its capability to generalize across all attack vectors present in the CICIDS 2017 dataset.

Proposed Model

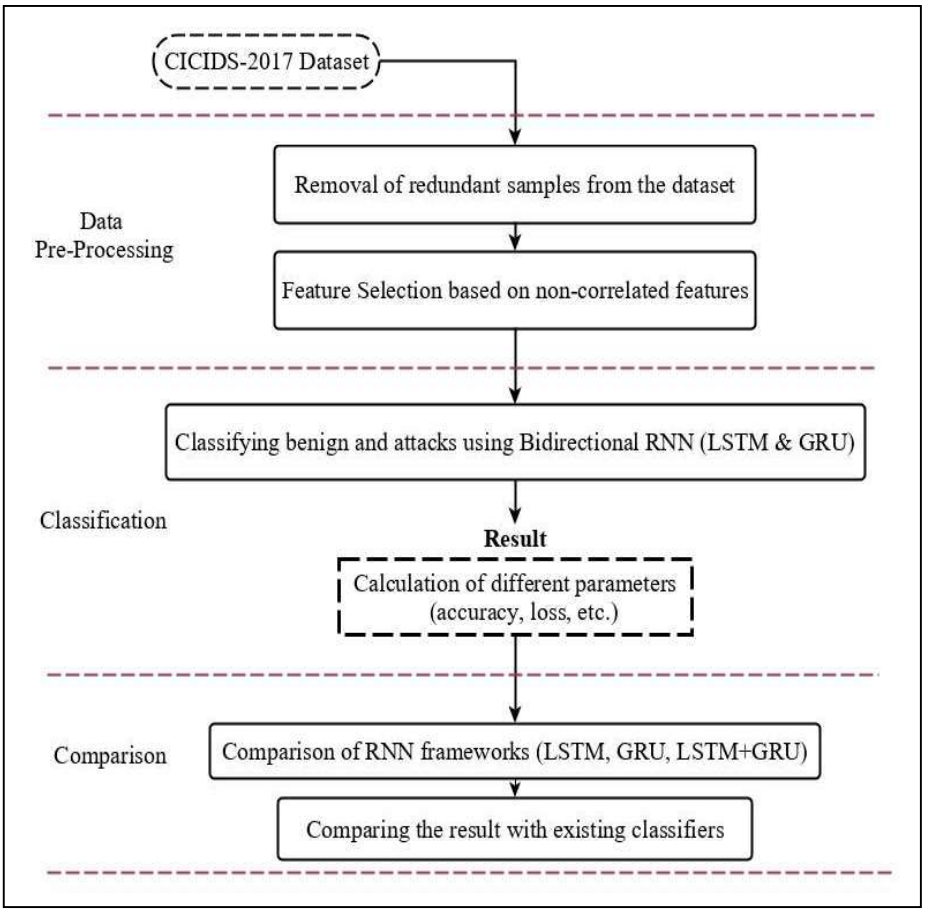

Figure 1: Proposed Model layout

In this study instead of using a single directional neural network layer, we have used a bidirectional RNN approach. In Bidirectional RNN there are two layers piled high on top of one another. A single layer moves in the forward direction while the other layer moves in the backward direction. The layer from behind does not interact with the forward layer and is solely attached to the output layer. The output of both layers is then combined using operations such

as concatenate, sum, multiply, etc. In our study, we have used one LSTM sequence for forward pass and one GRU sequence for a backward pass. We have combined the outputs of both sequences using concatenate operation. Figure 2 shows the structure of Bidirectional (Ullah et al., 2018) (Mulder et al., 2015).

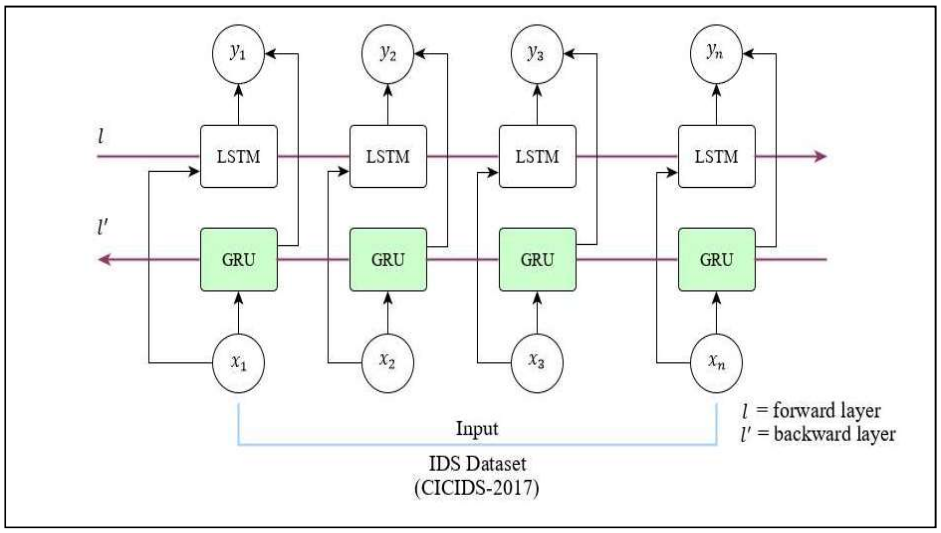

Figure 2. Bidirectional RNN Architecture

Algorithm 1 Proposed Bidirectional RNN

Input: data instances
Output: Confusion Matrix $\mathbf{C_{m\times m}}$
1. data pre-processing
2. reduction of duplicate samples from the dataset
3. computing correlation of the features set **C** using Equation (10)
4. creating Correlated Feature Set $\mathbf{C_f}$
5. if correlation value > 0.8
6. add feature to $\mathbf{C_f}$
7. else
8. increment in original feature set **C**
9. return $\mathbf{C_f}$
10. training Bidirectional RNN model with training samples
11. applying model to testing samples
12. return Confusion Matrix $\mathbf{C_{m\times m}}$
13. calculate Accuracy, Precision, Recall, TPR, and FPR

The algorithm 1 starts with the data pre-processing step that includes reading the data, separating the target class from the feature set, and separating the dataset into two fragments namely testing and training samples. The second step states the removal of duplicate or unwanted samples from the data. Then, correlated features are recognized using Pearson's Correlation Coefficient and dropped from the original feature set. Finally, benign activity and attacks are classified using the proposed model. The performance assessment is carried out using different parameters discussed in the latter section.

Result & Discussion

The experiments were performed with the Python language in the Google Colab Cloud Environment. The outcome of the tests is the confusion matrix of datasets. The CICIDS 2017 dataset is split into 67% of the training and 33% of the testing set.

CICIDS-2017 dataset contains 77 features and around two million samples. Removing correlated features and redundant samples from the dataset al-lowed the model to work more efficiently. The metrics of performance are calculated from the confusion matrix. The rows of the matrix represent the actual samples of classes and the column represents the predicted samples of the classes. The metric, True Positive (TP) of the Tuesday dataset is the values located diagonally in the matrix i.e. (137097, 1917, and 1055). More samples as True Positive show that the model has a high classification rate. Whereas True negative samples are samples inaccurately classified by the model.

Tuesday sub-dataset shows the prediction of FTP-Patator and SSH-Patator attacks. It has been noted that model has achieved a high TP rate and precision value for each class. Wednesday sub-dataset is the largest sub-dataset of the original CICIDS 2017 dataset. It also has the maximum number of attacks compared to other sub-datasets. The attacks predicted are GoldenEye, Hulk, Slowhttptest, slowloris, and Heartbleed. It is observed that the attack Hulk shows the highest true positive rate while the model does not correctly predict any Heartbleed attack samples. Figures 3 and 4 are the Confusion Matrix of Tuesday and Wednesday sub datasets. Thursday morning sub-dataset shows the prediction of Brute Force, SQL Injection, and XSS attacks. Brute Force attack is more accurately predicted than SQL Injection and XSS attacks. These reports show the prediction of attacks such as Infiltration, Bot, DDoS, and PortScan attacks. It is observed that the model does well while predicting Bot, DDoS, and PortScan. However, it does not perform well while predicting the Infiltration attack.

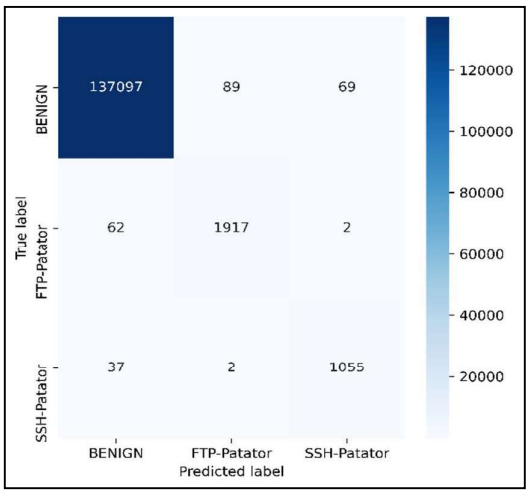

Figure 3: Tuesday: Confusion Matrix data set

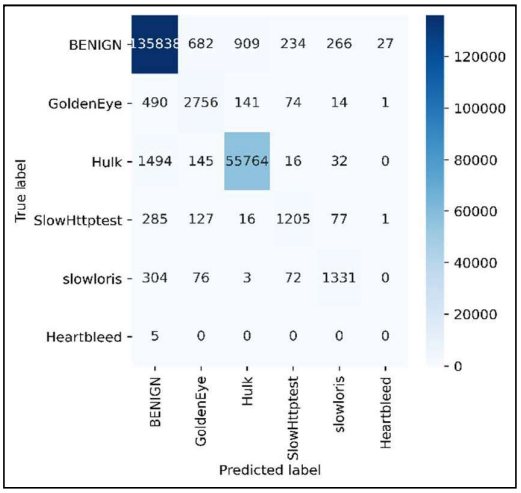

Figure 4: Wednesday Confusion Matrix data set

Table 1: Performance of Models

Research	No. of Features	Accuracy (%)
Ensemble classifier (Zhou Y. 2020) (Random Forest, Decision Tree, ForestPA)	-	97.3
KODE (Jaw E. 2021) K-means, SVM, DBSCAN	-	98
Random Forest	77	99.90
Naïve Bayes	77	64.49
K-Nearest Neighbour	77	99.67
Bidirectional RNN (LSTM + GRU)	<44	**99.13**
Bidirectional RNN (LSTM)	<44	**98.85**
Bidirectional RNN (GRU)	<44	**99.00**

The accuracy of each sub-dataset is calculated using multiple epoch settings considering the training and validation data loss. It is observed that the model has successfully achieved more than 97% accuracy for all sub-datasets. Accuracy of Thursday afternoon sub-dataset is the highest among all other subsets. Furthermore, we have contrasted the suggested model with different machine learning classifiers such as Random Forest, Naïve Bayes, K-Nearest Neighbour (KNN), the value of k is 9, and Ensemble method (Random Forest and KNN) in table 1. The proposed model Bidirectional RNN manages to score 99.13% of classification accuracy with nearly half (i.e. ≈ 58%) of the total features than other classifiers. The False Positive rate (FPR) of the proposed model is 0.074 which is less than the Naïve Bayes and KNN classifiers.

Conclusion

This study's proposal was a model based on deep learning to develop an Intrusion Detection System. We have used RNN with LSTM and GRU frameworks which re-solves the longer dependency issue of RNN. We have used one LSTM sequence and one GRU sequence with directions opposite to each other instead of using a single directional layer structure. The outputs of these two sequence layers are then com-bined using concatenate merge mode. We tested different combinations of LSTM and GRU. In addition, the different directions of LSTM and GRU sequences were also tested. However, the present strategy resulted in better performance. Different opti-mizers were explored to select the best optimizer for the model. Adaptive Gradient (AdaGrad) proved to be suitable for our proposed model. The suggested model was verified via the CICIDS-2017 dataset. Evaluation of the proposed model was done using performance metrics derived from the confusion matrix. It was observed that the suggested model functioned admirably in predicting the attacks. The proposed model scored classification accu-racy of 99.13% with a very low False Positive rate. The suggested design was com-pared with other classifiers such as Random Forest, Naïve Bayes, k-Nearest Neigh-bour (KNN), and Ensemble method. The precision of the suggested model was less than

Random Forest, KNN, and ensemble method but higher than the Naïve Bayes classifier. However, our model used nearly half the number (i.e., less than 44 features for each sub-datasets) of features when compared to all other classifiers. Moreover, we have also tested LSTM and GRU frameworks separately with RNN which re-sulted in GRU performing better than LSTM in terms of accuracy. Although the proposed approach has shown encouraging performance, it can be improved by further optimizing the classifier. The proposed model cannot effectively classify a few attacks. Therefore, the following proposed methodology is designed to overcome the issues we have faces in the previous model.

CHAPTER 3: COMPOSITION OF HYBRID DEEP LEARNING MODEL AND FEATURE OPTIMIZATION FOR INTRUSION DETECTION SYSTEM

We use a deep learning algorithm called CNN with GRU framework to develop an IDS. CNN is widely used as a deep learning technique in various fields. However, these methods have a problem with long-term dependencies. The existing deep learning solutions deal with this problem of long-term dependencies. When long-term interactions are given exponentially decreasing weights, the long-term dependency problem or problem occurs. Consequently, it becomes challenging for the model to track previous data. GRU can fix this issue and enhance the deep learning model. Hence to effectively detect the attacks we have used the GRU framework with CNN. To effectively classify the attacks, we have designed an IDS using a deep learning technique. We have used CNN to detect the attacks in IDS. To overcome the limitations of the existing DL technique as discussed in the previous section, the technique is modified using the GRU framework. We have tried different combinations of CNN layers and GRU sequences. However, we achieved better performance with three CNN layers and two GRU layers. In addition, we have optimized the CICIDS 2017 dataset for better performance. The optimization includes the removal of redundant features and the selection of distinguished features.

Motivation

To address the above discussed challenges, we developed an IDS using a hybrid model of Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). This approach, detailed in my published work, demonstrated high accuracy and reduced false positives with fewer features. However, it failed to detect certain attack types, highlighting a gap in its capability to generalize across all attack vectors present in the CICIDS 2017 dataset. To overcome this limitation, I further refined the IDS by integrating Convolutional Neural Networks (CNN) with GRU, aiming to leverage CNN's strength in feature extraction and GRU's efficiency in handling sequential data. This method showed improved performance over traditional CNN models but still left a few attack types undetected, as detailed in my subsequent publication.

Proposed Model

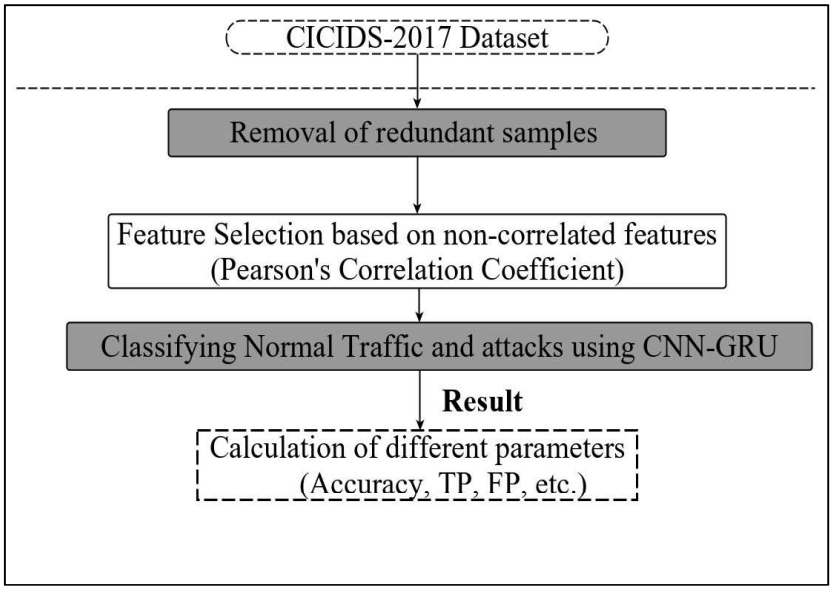

Figure 5. Proposed Model Layout

This segment manifests the flow of our proposed work. Proposed work involves two sections, namely feature optimization technique and the CNN-GRU model. Data pre-processing and classification are the two processes that make up the entire workflow in Figure 5. The elimination of repetition and choosing the ideal feature set are part of the first stage. The second stage includes categorising the data and computing several parameters. The assessment of the technique's performance in comparison to other current algorithms is shown in the subsequent part.

Algorithm 2 Feature Set Optimization

Input: data instances
1. reduction of duplicate samples from the dataset
2. computing correlation of the features setCusing Pearson's Correlation equation
3. creating Correlated Feature Set C_f
4. if correlation value >0.8
5. add feature to C_f
6. else
7. increment in original feature setC
8. return C_f

Algorithm 3 CNN-GRU

Input: data instances
1. reduction of duplicate samples from the dataset (Algorithm 1)
2. computing correlation of the features set C using Pearson's Correlation equation (Algorithm 1)
3. training CNN-GRU model with training samples
4. applying model to testing samples
5. return Confusion Matrix $C_{(m*m)}$

Result & Discussion

Table 2: Performance of Models

Model	No. of attributes	Accuracy
Proposed CNN-GRU	Less than 44	98.73%
CNN (Yang and Wang, 2019)	77	94.96%
CNN (Ho et al., 2021)	77	99.7%
CNN-LSTM (Priyanka and Gireesh Kumar, 2022)	77	98.67%
CNN (Syed et al., 2020)	77	99.47%

Multiple epoch settings are used to calculate each sub-accuracy dataset while taking the loss of training/validation data into account. For all subsets, it has been noted that the model has effectively attained greater than 95% accuracy. The Tuesday sub-dataset has the highest accuracy compared to the other subsets. Moreover, we have also compared the proposed model with other Convolution Neural Network models. It has been reported that the suggested model CNN-GRU achieves a classification accuracy score of 98.73% while using just about half (i.e., 58%) of the total features compared to other models. The proposed model can achieve this accuracy for the full CICIDS 2017 dataset. It also detects attacks such as SQL injection and SSH Patator which other model fails to detect. However, it fails to detect Heartbleed and XSS attacks.

Table 2 shows the assessment of the proposed technique with other existing CNN models. The CNN-GRU model manages to score a better accuracy rate than models developed by Samson H. et al. (2020) and Sun P. et al (2020). The other two CNN models (Priyanka V. et al. 2021) and (Maseer Z. et al. 2021) have slightly high accuracy scores than our proposed technique. However, these two models were evaluated on the partial dataset. Moreover, we have analysed the classification time for each sub-datasets against their sizes and accuracies respectively. We have evaluated the techniques using different epochs ranging from 20 to 100 for different sub-datasets.

Conclusion

In this research, we designed a DL technique to develop an IDS. We have used CNN with the GRU framework. We have used three CNN layers and two GRU sequences. Both sequences use an activation function called ReLU to define the output of the given inputs. Following that, the concatenate merge mode is used to combine the outputs of both sequences. The dataset considered in this work offers a diversity of attacks and a high number of instances, which was used to verify the suggested model. The dataset contains network data recorded from eight different sessions. In this study, we have considered seven sessions and overlooked one session which contains just normal traffic data. We optimized the original dataset by selecting features based on Pearson's Correlation coefficient. We also removed redundant or duplicate instances from the dataset. The evaluation of the model was done using measurement parameters of the

confusion matrix. The suggested model achieved a low FP rate and a classification accuracy rate of 98.73%. The technique used nearly half of the total attributes (less than 58% of the total for each sub-dataset) when compared to all other classifiers. Also, we have observed the uneven distribution of the samples in the dataset which do not allow the model to detect a few attacks in the dataset. Therefore, to overcome this limitation we have designed the approach which oversamples the samples before passing them to the training phase. The same is discussed in the next section.

CHAPTER 4: ASSESSING THE EFFECTIVENESS OF GRU FOR DEEP LEARNING-BASED INTRUSION DETECTION

Chapter 4 focuses on evaluating the performance of Gated Recurrent Units (GRU) in deep learning-based intrusion detection systems. The motivation for this chapter stems from the need to address specific challenges in network security, particularly the detection of complex and evolving cyber threats. While traditional machine learning approaches have demonstrated utility in analyzing network traffic, their inability to effectively capture temporal dependencies limits their application in real-world scenarios.

GRU, as a variant of Recurrent Neural Networks (RNN), has shown significant promise in addressing these challenges due to its ability to manage sequential data efficiently. Unlike standard RNNs, GRUs mitigate the vanishing gradient problem, enabling the analysis of long-term dependencies in network traffic. This capability is particularly crucial for detecting sophisticated attack patterns that unfold over extended periods.

The research presented in this chapter aims to:

1. Investigate the suitability of GRU for intrusion detection by analyzing its architecture and operational advantages.
2. Examine the role of GRU in capturing temporal dependencies within network traffic data to improve detection accuracy.
3. Evaluate the effectiveness of GRU in comparison to other deep learning models, such as LSTM and traditional RNNs, through rigorous experimentation.

Furthermore, the chapter explores the integration of GRU with advanced data preprocessing techniques, such as feature selection and dataset balancing, to optimize its performance. By focusing on GRU, the research aims to highlight its potential as a robust solution for developing scalable and adaptive intrusion detection systems capable of addressing the dynamic nature of cyber threats.

Proposed Method

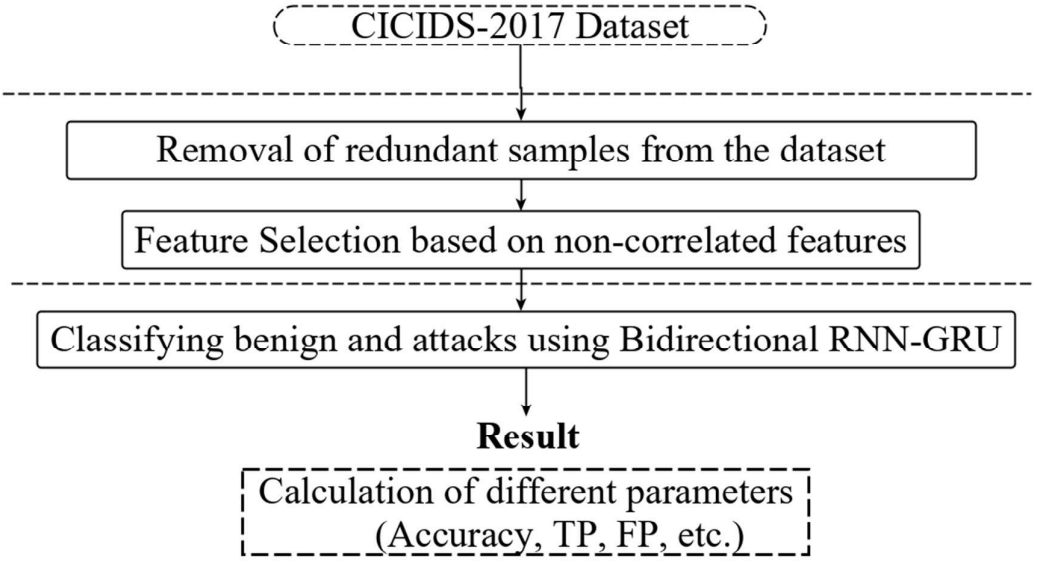
CICIDS-2017 Dataset
Removal of redundant samples from the dataset
Feature Selection based on non-correlated features
Classifying benign and attacks using Bidirectional RNN-GRU
Result
Calculation of different parameters
(Accuracy, TP, FP, etc.)

Figure 6. Proposed Model Layout

In this study, we have used a dataset created by a research crew in the Canadian Institute of Cybersecurity i.e. CICIDS 2017. This dataset covers more variants of features and metadata compared to traditional datasets. However, to remove the redundancy in the dataset and to find the optimum feature subset, pre-processing is performed. The filter method called Pearson's Correlation Coefficient finds the optimum subgroup of attributes from the attribute set. It measures the degree of linear correlation or similarity of the features in the dataset and delivers a correlation coefficient value that ranges between -1 to 1 i.e. [-1, 1] (Zhu et al., 2019) (Feng et al., 2018).

This study showcase a model using a neural network algorithm called Recurrent Neural Network (RNN). RNN provides the ability to resolve issues such as long term dependencies and variable correlations that are seen in other neural networks. RNN enables the network to add feedback on the outputs of the previous time step in the current time step. Gated Recurrent Unit (GRU) framework is designed to combat exploding or vanishing gradient problems. GRU is the improved version of the LSTM framework. GRUs has a gate structure having two gates namely the reset and update gates (Rezk et al., 2020b). We have used a bidirectional RNN where two GRU layers are stacked on top of each other. Both layers are kept in opposite directions. In this proposed method, GRU layer moves in the forward direction whereas the other GRU layer moves in the backward direction.

TABLE 3. Accuracy

Sub-Dataset	Accuracy (%)
Tuesday	0.9984
Wednesday	0.9676
Thursday Morning	0.9923
Thursday Afternoon	0.9995
Friday Morning	0.9923
Friday Afternoon-DDoS	0.9847
Friday Afternoon-PortScan	0.9951

TABLE 4. Statistical parameters of each attacks

Attacks	TPR	FPR	Precision
FTP-Patator	0.9846	0.0002	0.9817
SSH-Patator	0.9216	0.0004	0.9398
Goldeneye	0.8394	0.0045	0.7593
Hulk	0.9615	0.0136	0.9652
Slowhttptest	0.7515	0.0028	0.6912
Slowloris	0.7886	0.0030	0.6947
Heartbleed	0	0.0008	0
Brute Force	0.5933	0.0032	0.6317
Sql Inj	0.5446	0.0033	0.4
XSS	0.25	0.0002	0.0666
Infiltration	0	0.0002	0
Bot	0.6172	0.0036	0.6354
DDoS	0.9809	0.0101	0.9923
PortScan	0.9934	0.0036	0.9950

The experiments were performed using the python programming language on the Google Colab platform. Python libraries such as sklearn, keras, tensorflow were used to classify and analyse the data using the proposed model. To test the proposed model, the CICIDS 2017 dataset is divided into training and testing sets i.e. 67% - training set and 33% - testing set. As discussed in the previous section, the correlation-based feature selection technique reduced the feature set to less or equal to 43 features for all sub-datasets. Table 4 shows the TPR, FPR, and precision values for all the attacks in the dataset. The results show that most of the attacks are classified with higher accuracy. However, attacks such as Heartbleed and XSS are not predicted accurately by the proposed model. One of the reasons behind this is the lesser amount of training data for these attacks. Moreover, the results show very less FPR value for most of the attacks. As a result, the precision values are higher.

Table 3 shows the accuracy values for each sub-dataset of the CICIDS 2017 dataset. The results show that the proposed model gives more than 96% accurate predictions for each sub-datasets. Thursday-Afternoon sub-dataset which contain Infiltration attacks shows the highest accuracy value among the other sub-datasets. Our proposed model has an average accuracy value of 98.99%. Table 5 shows the comparison of our proposed model with other related approaches for IDS. Naïve Bayes is a machine learning algorithm based on probability. Along with the Bidirectional RNN-GRU model we have also tested the Naïve Bayes classifier. However, it shows the least accuracy among other classifiers. Moreover, the proposed model is also

compared with other techniques namely Random Forest (Pelletier et al., 2019), Artificial Neural Network (Pelletier et al., 2019) and AdaBoost based IDS (Yulianto et al., 2019). These techniques are also evaluated on CICIDS 2017 dataset. Our proposed model outperforms these techniques with the highest accuracy.

TABLE 5. Performance of classifiers

Classifier	Accuracy
Naïve Bayes	64.49%
Random Forest (Pelletier et al., 2019)	96.24%
ANN(50 iterations) (Pelletier et al., 2019)	87.79%
ANN(500 iterations) (Pelletier et al., 2019)	96.53%
AdaBoost-based IDS (Yulianto et al., 2019)	81.83%
Proposed Bidirectional RNN-GRU	**98.99%**

Conclusion

In this study, we introduced the design and development of an Intrusion Detection System (IDS) leveraging a deep learning approach, specifically employing the Recurrent Neural Network (RNN) with Gated Recurrent Units (GRU) framework. While existing research in the domain of IDS has explored both traditional machine learning methods and advanced deep learning techniques, significant challenges persist. Among these challenges, the inability to handle inefficient or imbalanced data effectively and the difficulty in managing long dependencies within sequential data remain critical obstacles. These limitations often lead to suboptimal detection accuracy, particularly when addressing complex or evolving attack patterns in real-world scenarios. This underscores the urgent need for innovative solutions and further research in the field to overcome these barriers. The primary objective of this chapter was to evaluate the potential of GRU in addressing the long dependency problem that frequently hinders the performance of other deep learning techniques. GRUs, with their ability to manage sequential data more effectively, allow the model to retain critical information over extended periods without succumbing to the vanishing gradient problem. Our proposed RNN-GRU framework successfully tackled this issue, achieving an impressive accuracy of 98.99%, a significant improvement over many existing models.

CHAPTER 5: ENHANCING DATASET FOR OPTIMIZED CLASSIFICATION IN INTRUSION DETECTION SYSTEM

Motivation

To address the above discussed challenges, we developed an IDS using a hybrid model of Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). This approach, detailed in my published work, demonstrated high accuracy and reduced false positives with fewer features. However, it failed to detect certain attack types, highlighting a gap in its capability to generalize across all attack vectors present

in the CICIDS 2017 dataset which made us to refine the IDS by integrating Convolutional Neural Networks (CNN) with GRU, aiming to leverage CNN's strength in feature extraction and GRU's efficiency in handling sequential data. This method showed improved performance over traditional CNN models but still left a few attack types undetected. Recognizing the need for a more balanced and comprehensive detection approach, I incorporated the Synthetic Minority Over-sampling Technique (SMOTE) to address the class imbalance in the dataset. This final approach successfully detected all attack types in the dataset, achieving a more robust and reliable IDS.

Proposed Model

Multiple datasets are available for Intrusion Detection System, and the Canadian Institute of Cybersecurity IDS dataset (CICIDS) is one among these. A well-designed dataset is crucial for creating a dependable IDS. This requires taking into account factors such as the diversity of traffic, the volume of samples, and varied attacks. This dataset was used in this research to assess the model's performance. The dataset is up-to-date, inclusive of various features and metadata, and more comprehensive than previous datasets. It contains results from 8 periods that collected data from two networks, the Victim Network and Attack Network. The initial session's outcome includes only benign traffic data, while the dataset features seven types of attacks that satisfy real-time traffic requirements. This dataset comprises 77 features in each subset, and in this study, redundant data were eliminated from the dataset.

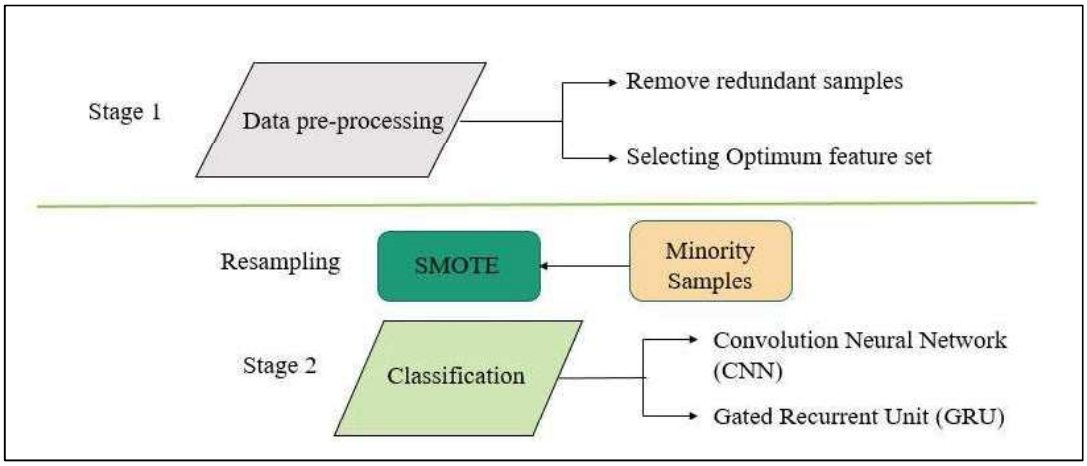

Figure 7: Proposed Model. SMOTE: Synthetic Minority Oversampling Technique

This is a hybrid oversampling technique that combines under-sampling and oversampling. It first under-samples the majority class to remove noisy samples and then oversamples the classes with less instances to balance the classes. SMOTE is a popular oversampling method used in ML to address the issue of class imbalance. This technique is used when the number of samples belonging to the minority class is much lower compared to the majority class, resulting in a bias towards the majority class in a predictive model. The basic idea behind SMOTE is to generate synthetic samples for the minority class by interpolating between the existing samples. For each minority sample, SMOTE selects k nearest neighbours and generates synthetic

samples by computing the linear interpolation between the feature values of the sample and its neighbours. The newly generated samples are added to the original minority class data to balance the class distribution. SMOTE helps to improve the performance of ML algorithms on imbalanced datasets by generating synthetic samples for the minority class. However, it is significant to keep in mind that oversampling can lead to overfitting, and it is recommended to use techniques such as cross-validation and ensemble learning to avoid overfitting. In this research, we have used this technique to balance the attack data of the CICIDS 2017 dataset. The samples in the dataset are highly imbalanced. Therefore, to train the model perfectly we have optimized the dataset. We have trained the CNN-GRU model with the optimized dataset and compared it with the previous results. This steps of this methodology are discussed in the algorithm 4.

Algorithm 4 Dataset balancing and classification

Input: Dataset
Output: Balanced dataset
1. Confusion matrix
2. Data enhancement
3. Remove the redundancy
4. Feature Optimization
5. Using Pearson's Correlation formula, calculate the correlation of the feature set C.
6. if correlation_value ¿ 0.8
7. include feature to C_f
8. else
9. increment in feature set C
10. return C_f
11. Dataset Balancing using SMOTE
12. Calculating instances of each classes
13. Generating instances for classes with less instances.
14. Samples are generated until all the classes have equal amout of samples.
15. Classification
16. Classification on balanced dataset.
17. return Confusion Matrix C_{m*m}
18. Comparing the results
19. End

Result & Discussion

The optimized dataset i.e. after balancing the samples of the dataset, the DL model CNN-GRU is trained with the same. We have used Google Colab and python language to execute the experiments. The previous outcomes of the model are compared with that of the model using an optimized or balanced dataset. The first part of this section shows the comparison of the samples before and after the oversampling was performed. It is observed that the technique managed to oversample or balance the samples for all the classes in the dataset. The Tuesday sub-dataset had 415785, 6095, and 3360 samples for three classes before the technique was applied. Now, the SMOTE technique has oversampled the samples to 415785 for all three classes. Similarly, samples for all the sub-datasets are oversampled considering the highest samples.

TABLE 6 Wednesday Sub-dataset

Wednesday Sub-dataset		
	Imbalanced	Balanced
BENIGN	418569	418569
Goldeneye	10286	418569
Hulk	173794	418569
Slowhttptest	5242	418569
Slowloris	5385	418569
Heartbleed	11	418569

TABLE 7 Thursday Morning Sub-dataset

Thursday Morning Sub-dataset		
	Imbalanced	Balanced
BENIGN	162157	162157
Brute Force	1470	162157
Sql Inj	652	162157
XSS	21	162157

TABLE 8 Performance comparison

Attacks	**With imbalanced dataset**			**With balanced dataset**		
	TPR	**TP**	**TN**	**TPR**	**TP**	**TN**
benign	0.99961371	137150	2919	0.98284925	134441	274100
ftpPatatorAttack	0.9754509	1947	138292	0.99601151	137097	272825
sshPatatorAttack	0.85587975	968	139184	0.99198222	136095	272336
benign	0.97300994	134433	59203	0.92272685	127734	680067
goldeneyeAttack	0.77670753	2661	198336	0.93574067	128946	678562
hulkAttack	0.93301018	53510	142083	0.96166488	132829	686748
slowhttptestAttack	0.73820755	1252	200466	0.92944111	127787	680332
slowlorisAttack	0.82234957	1435	200366	0.91475848	126598	681578
heartbleedAttack	0	0	202377	1	138528	690203
benign	0.99957035	53509	622	0.97710695	52242	160233
bruteForceAttack	0.88912134	425	53539	0.13131332	7064	159262
SqlInjectionAttack	0.02970297	6	54005	0.98878199	52621	116189
XSSAttack	0	0	54212	1	53569	157908

benign	0.99963104	83989	1	0.98424072	82690	82980
infiltrationAttack	0.14285714	1	83989	0.98768077	82980	82690
benign	0.9998669	60096	0	0.96751975	58295	59534
botAttack	0	0	60096	0.99231603	59534	58295
benign	0.95862989	30170	42041	0.97826241	41313	41132
ddosAttack	0.99298503	42041	30170	0.97321598	41132	41313
benign	0.98890902	40391	29592	0.99476248	40645	40173
portscanAttack	0.99255383	29592	40391	0.99153421	40173	40645

As shown in table 8, the performance of the method has increased after the optimization of the dataset samples. It is observed that the attacks such as heartbleed, XSS, and bot remained undetected in the previous research work. Nevertheless, the same attacks are detected with a decent true positive rate after the samples were balanced in the current research work. Zero samples were correctly classified for these attacks in the previous work whereas, in this work the TPR is 1, 1, and 0.99 for heartbleed, XSS, and bot respectively. Moreover, the TPR for SQL injection and infiltration attacks is exponentially increased after the optimization. The average TPR of the model before applying the balancing technique is 71.75%, which was enhanced further to 93.34%. Hence, it is observed that the proposed technique improves the previous model with a big difference.

Conclusion

In this research work we have proposed a method to efficiently detect all the attacks while implementing the DL model. The dataset we have used is CICIDS 2017. This dataset has imbalanced samples for the attacks. As a result the detection of the some attacks is inaccurate. Therefore, we have optimized the dataset with the oversampling technique called SMOTE. Moreover, we have compared the performance of the DL model CNN-GRU before and after the dataset was optimized. It has been observed that the technique managed to enhance the performance of the method. The attacks which were previously undetected or inaccurately detected are being detected with a better true positive rate.

CHAPTER 6: CONCLUSION & FUTURE SCOPE

The development of robust intrusion detection systems (IDS) is essential for safeguarding digital infrastructures against an ever-increasing array of sophisticated cyber threats. This study focused on addressing the limitations of current IDS approaches through the application of advanced deep learning techniques and innovative data preprocessing methods. Initially, a comprehensive review of the state-of-the-art IDS methods, including ensemble learning, federated learning, statistical flow features, and deep learning approaches, was conducted. This review laid the groundwork for identifying the potential and shortcomings of existing techniques, highlighting the need for more effective and adaptable IDS solutions.

To address these gaps, we first developed an IDS using a combination of Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). This model achieved high accuracy and low false positive rates with fewer features, demonstrating the efficacy of deep learning in anomaly detection. However, it fell short in detecting certain attack types, pointing to the need for further refinement.

In response, we introduced a second IDS model combining Convolutional Neural Networks (CNN) with GRU. This hybrid approach leveraged CNN's powerful feature extraction capabilities alongside GRU's efficiency in handling sequential data. While this model outperformed existing CNN models, it still failed to detect a few attack types, underscoring the ongoing challenge of comprehensive threat detection. To overcome this limitation, we employed the Synthetic Minority Over-sampling Technique (SMOTE) to address the imbalanced distribution of attack types in the dataset. This final approach successfully detected all attack types in the CICIDS 2017 dataset, demonstrating the critical importance of data preprocessing in enhancing IDS performance.

Overall, this study makes significant contributions to the field of IDS by developing and evaluating advanced deep learning-based models and emphasizing the role of data balancing techniques in achieving comprehensive threat detection. Our research underscores the need for continuous innovation and adaptation in IDS methodologies to keep pace with the dynamic and evolving nature of cyber threats. The journey from RNN-LSTM-GRU based IDS to CNN-GRU based IDS, and the integration of SMOTE, highlights the iterative process of improving IDS performance and reliability. This work lays a strong foundation for future research aimed at developing even more robust and effective IDS solutions.

Future Scope

There are a few extensions of this work which can be carried out in future to enhance this study further.

Online Learning and Incremental Updates: Online learning and incremental updates are essential components for the evolution of an Intrusion Detection System (IDS) in today's rapidly changing cybersecurity landscape. By implementing online learning techniques, an IDS can adapt and learn from new data as it arrives, enabling it to stay up-to-date with emerging threats and vulnerabilities. This approach allows the IDS to continuously refine its models and detection capabilities, making it more resilient against evolving attack strategies, including zero-day exploits. Incremental updates enable the system to incorporate new information without retraining the entire model, thus minimizing computational overhead and ensuring that the IDS remains effective in real-time or near real-time scenarios. As cyber threats constantly evolve, the ability to adapt on the fly and learn from the latest data is a critical feature for any modern intrusion detection system, enhancing its capacity to protect networks and systems from emerging threats.

Real-time Processing: Real-time processing is a pivotal facet of an effective Intrusion Detection System (IDS) that allows it to swiftly identify and respond to security threats as they occur. In today's hyper-connected digital environments, where milliseconds can make a significant difference, the ability to analyze and make decisions on network traffic in real-time

is paramount. Real-time processing empowers the IDS to swiftly detect anomalies or malicious activities, reducing the window of opportunity for attackers and enabling immediate countermeasures. This capability not only enhances the security posture but also minimizes potential damage and data breaches by providing timely alerts and responses. Furthermore, real-time processing enables the IDS to adapt to changing network conditions and traffic patterns dynamically, making it an indispensable component for safeguarding modern networks against a wide range of cyber threats.

Deployment and Testing in Real-world Scenarios: Deployment and testing of an Intrusion Detection System (IDS) in real-world scenarios are pivotal stages in its development and validation. While laboratory testing provides controlled environments for assessing performance, real-world deployment is where an IDS truly proves its worth. In live network environments, the system faces the complex challenges of diverse traffic patterns, evolving attack tactics, and the potential for false positives, all of which must be managed effectively to avoid disrupting legitimate operations. Moreover, real-world deployment allows for the evaluation of the IDS's practicality, scalability, and ability to seamlessly integrate with existing security infrastructures. It's in these environments that the system's response time, reliability, and overall effectiveness can be accurately measured, providing critical insights for further refinements and optimizations. Ultimately, deploying and testing an IDS in real-world scenarios is the litmus test that ensures its readiness to protect networks and systems from actual cyber threats and vulnerabilities.

Evaluation on Different Datasets: Evaluating an Intrusion Detection System (IDS) on different datasets is a crucial step in assessing its effectiveness and generalizability. While initial development and testing on a specific dataset, such as the CICIDS 2017 dataset, provide valuable insights, validating the IDS on diverse datasets from various sources is essential to ensure its robustness and real-world applicability. Different datasets may exhibit unique characteristics, traffic patterns, and attack scenarios, which challenge the IDS to adapt and perform consistently across varied environments. Evaluating the system on multiple datasets allows researchers and practitioners to gauge its ability to detect a broader spectrum of threats and vulnerabilities, ultimately enhancing its reliability and trustworthiness as a security solution. Moreover, it aids in uncovering potential weaknesses or biases in the IDS, facilitating improvements and adjustments to optimize its performance in a wider range of cybersecurity contexts.

LIST OF PUBLICATIONS

1. **Henry, A. A**., Gautam, S. K. Application of Big Data and Machine Learning. In: Dulhare, U. N., ed. Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications. John Wiley & Sons, 2020; 2020:305-336.
2. **Azriel, H.**, Sunil, G. Application of Machine Learning to Detect Phishing URLs. Research Magazine, Institute of Advanced Research, 2020.
3. **Gautam, S.**, Henry, A. A Composite Approach of Intrusion Detection Systems: Hybrid RNN and Correlation-Based Feature Optimization. Electronics, 2022, 11, 3529.

4. **Henry, A.**, Gautam, S., Khanna, S. Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. Sensors, 2023, 23, 890.
5. **Henry, A.**, Gautam, S. Intelligent Intrusion Detection System Using Deep Learning Technique. Computing, Communication and Learning, Springer Nature, 2022.
6. Gautam, S., **Henry, A. A.** Wireless Sensor Networks with the Context of Knowledge-Based Management. Studies in Computational Intelligence for Wireless Sensor, Springer, 2021.
7. **Henry, A. A.**, Gautam, S. K. The Future of Intrusion Detection Systems in Cloud Computing. Horizons in Computer Science Research, Nova Science, 2022.
8. **Henry, A. A**., Gautam, S. K., Khanna, S. O. Boosting Intrusion Detection System Performance Through Enhanced Dataset. Device Intelligence, Computing and Communication Technologies (DICCT-2025) (Communicated)
9. **Henry, A. A**., Gautam, S. K., Khanna, S. O. Exploring Advances in Anomaly- Based Intrusion Detection Systems: A Survey of Methodologies and Applications. Device Intelligence, Computing and Communication Technologies (DICCT-2025) (Communicated)

Bibliography

[1] Prasad, M.; Tripathi, S.; Dahal, K. An efficient feature selection-based Bayesian and Rough set approach for intrusion detection. Appl. Soft Comput. J. 2020, 87, 105980.
[2] Dutt, I.; Borah, S.; Maitra, I.K. Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model. IEEE Access 2020, 8, 34929–34941.
[3] Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity 2019, 2, 20.
[4] Sultana, N.; Chilamkurti, N.; Peng, W.; Alhadad, R. Survey on SDN based network intrusion detection system using ML approaches. Peer-Peer Netw. Appl. 2018, 12, 493–501.
[5] Jyothsna, V.; Prasad, V.V.R.; Prasad, K.M. A Review of Anomaly based Intrusion Detection Systems. Int. J. Comput. Appl. 2011, 28, 26–35.
[6] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access. 2019;7:41525-41550.
[7] Kwon D, Kim H, Kim J, Suh SC, Kim I, Kim KJ. A survey of deep learning-based network anomaly detection. Cluster Computing. 2017;22(S1):949-961.
[8] Fernandez GC, Xu S. A Case Study on using Deep Learning for Network Intrusion Detection. MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM). 2019.
[9] Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Computers & Security. 2019;86:147-167.
[10] Guo L, Li N, Jia F, Lei Y, Lin J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. Neurocomputing. 2017;240:98-109.

[11] Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid. 2019;10(1):841-851.
[12] Devarakonda N, Pamidi S, Kumari VV, Govardhan A. Intrusion Detection System using Bayesian Network and Hidden Markov Model. Procedia Technology. 2012;4:506-514.
[13] Sajjad SM, Bouk SH, Yousaf M. Neighbor Node Trust based Intrusion Detection Sys-tem for WSN. Procedia Computer Science. 2015;63:183-188.
[14] Belavagi MC, Muniyal B. Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. Procedia Computer Science. 2016;89:117-123.
[15] Gaikwad D, Thool RC. Intrusion Detection System Using Bagging with Partial Deci-sion TreeBase Classifier. Procedia Computer Science. 2015;49:92-98.
[16] Ahmad I, Basheri M, Iqbal MJ, Rahim A. Performance Comparison of Support Vec-tor Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. IEEE Access. 2018;6:33789-33795.
[17] Kasongo SM, Sun Y. A Deep Learning Method With Filter Based Feature Engineer-ing for Wireless Intrusion Detection System. IEEE Access. 2019;7:38597-38607.
[18] Yang H, Wang F. Wireless Network Intrusion Detection Based on Improved Con-volutional Neural Network. IEEE Access. 2019;7:64366-64374.
[19] Yin C, Zhu Y, Fei J, He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access. 2017;5:21954-21961.
[20] Kim A, Park M, Lee DH. AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection. IEEE Access. 2020;8:70245-70261.
[21] Wang M, Zheng K, Yang Y, Wang X. An Explainable Machine Learning Framework for Intrusion Detection Systems. IEEE Access. 2020;8:73127-73141.
[22] Hao Y, Sheng Y, Wang J. Variant Gated Recurrent Units With Encoders to Preprocess Packets for Payload-Aware Intrusion Detection. IEEE Access. 2019;7:49985-49998.
[23] Ullah A, Ahmad J, Muhammad K, Sajjad M, Baik SW. Action Recognition in Video Se-quences using Deep Bi-Directional LSTM With CNN Features. IEEE Access. 2018;6:1155-1166.
[24] Mulder WD, Bethard S, Moens M-F. A survey on the application of recurrent neural networks to statistical language modeling. Computer Speech & Language. 2015;30(1):61-98.
[25] Magán-Carrión, R., Urda, D., Díaz-Cano, I., & Dorronsoro, B. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. Applied Sciences, 10(5), 1775.
[26] Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020). A machine learning approach for intrusion detection system on NSL-KDD dataset. 2020 International Conference on Smart Electronics and Communication (ICOSEC).
[27] Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet, 13(5), 111.
[28] Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini impurity-based weighted random forest (GIWRF) feature selection technique. Cybersecurity, 5(1).
[29] De Carvalho Bertoli, G., Pereira Junior, L. A., Saotome, O., Dos Santos, A. L., Verri, F. A., Marcondes, C. A., Barbieri, S., Rodrigues, M. S., & Parente De Oliveira, J. M. (2021).

An end-to-end framework for machine learning-based network Intrusion Detection System. IEEE Access, 9, 106790–106805.
[30] Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for Intrusion Detection Systems. IEEE Access, 8, 73127–73141.
[31] Ho S, Jufout S Al, Dajani K, Mozumdar M. A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network. IEEE Open J Comput Soc. 2021;2(February):14-25.
[32] Priyanka, V., Gireesh Kumar, T. (2022). Performance Assessment of IDS Based on CICIDS-2017 Dataset. In: Joshi, A., Mahmud, M., Ragel, R.G., Thakur, N.V. (eds) Information and Communication Technology for Competitive Strategies (ICTCS 2020). Lecture Notes in Networks and Systems, vol 191. Springer, Singapore.
[33] Sun P, Liu P, Li Q, et al. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Secur Commun Networks. 2020;2020.
[34] Dong, S., Xia, Y., & Peng, T. (2021). Network abnormal traffic detection model based on semi-supervised Deep Reinforcement Learning. IEEE Transactions on Network and Service Management, 18(4), 4197–4212.
[35] Injadat, M. N., Moubayed, A., Nassif, A. B., & Shami, A. (2021). Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management, 18(2), 1803–1816.
[36] Sheba K, Raj SG. An approach for automatic lesion detection in mammograms. Cogent Engineering. 2018;5(1).
[37] Srinivasa, K.G., et al. Evaluating ML approaches for intrusion detection using the CSE-CIC-IDS2018 dataset. Journal of Network and Computer Applications, 2020, 152, 102505.
[38] Li, Q., Liang, F., Xu, D., Zhang, C., & Wang, S. Investigating the Effect of Partial Balancing Techniques on Classification Performance of Imbalanced Datasets. IEEE Access, 2019, 7, 99703-99715
[39] Balla, A., Hartman, S., & Lippmann, R. Impact of imbalanced data sets on the performance of intrusion detection systems for supervisory control and data acquisition systems. In 2018 IEEE International Conference on Intelligence and Security Informatics ISI, IEEE, 2018, 199-204.
[40] Liu, I., Xu, Y., & Li, K. A GAN-based data balancing algorithm for robust network intrusion detection. Journal of Network and Computer Applications, 130, 2019, 120-129.
[41] Priyanka, et al. Comprehensive Evaluation of Intrusion Detection Systems using CICIDS-2017 Dataset. International Journal of Advanced Science and Technology, 2021, 30(4), 15137-15147.
[42] Jose, A., Viswanath, G., & Nair, A. Evaluation of DL Algorithms for Intrusion Detection in IoT using CICIDS-2017 Dataset. In 2019 International Conference on Communication and Signal Processing (ICCSP), IEEE, 2019, 1230-1234.
[43] Lanvin, J., Josse, G., & Jourjon, G. Investigating errors in intrusion detection datasets: the case of the CICIDS2017 dataset. In Proceedings of the 34th Annual ACM Symposium on Applied Computing, 2019, 204-211.
[44] Stiawan, D.; Idris, M.Y.; Bamhdi, A.M.; Budiarto, R. CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection. IEEE Access 2020, 8, 132911–132921.

[45] Zhu, H.; You, X.; Liu, S. Multiple Ant Colony Optimization Based on Pearson Correlation Coefficient. IEEE Access 2019, 7, 61628–61638
[46] Chawla, N. V., & Gionis, A. (2021). SMOTE-X: A Synthetic Minority Over-sampling Technique Using eXplanations. IEEE Transactions on Knowledge and Data Engineering, 33(3), 2021, 982-995.
[47] Ding, L., Yang, J., & Zhou, Y. Combining SMOTE with clustering algorithm for imbalanced classification. Journal of Ambient Intelligence and Humanized Computing, 13(4), 2022, 4099-4109.